



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/824,251	04/13/2004	Andreas W. Wendker	APLE.P0004C	6431
62224	7590	01/24/2008		
ADELI & TOLLEN, LLP 1875 CENTURY PARK EAST SUITE 1360 LOS ANGELES, CA 90067			EXAMINER MYINT, DENNIS Y	
			ART UNIT	PAPER NUMBER
			2162	
			MAIL DATE	DELIVERY MODE
			01/24/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/824,251

Applicant(s)

WENDKER ET AL.

Examiner

Dennis Myint

Art Unit

2162

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 11/02/2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 20-27, 31, 32, 35-38, 42-44, 46-53 and 56-74 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

- 5) ☐ Claim(s) _____ is/are allowed.

- 6) ☒ Claim(s) 20-27, 31, 32, 35-38, 42-44, 46-53 and 56-63 is/are rejected.

- 7) ☐ Claim(s) _____ is/are objected to.

- 8) ☐ Claim(s) 64-74 are subject to restriction and/or election requirement.

Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action, or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This communication is responsive to Applicant's Amendment, filed on November 2, 2007.
2. Claims 20-27, 31-32, 35-38, 42-44, 46-53 and 56-74 are currently pending in this application. Claims 20, 31, 35, 42, 46, 52, 58, 59, 64, 66, 68, 69, 72, and 73 are independent claims. In the Amendment filed on November 2, 2007, Claims 26, 52, 59 and 60 were amended. Claims 64-74 were newly added. Claims 28-30, 33-34, 39-41, 45, and 54-55 were cancelled. **This office action is made final.**
3. In light of the cancellation of claims 34 and 45, rejection of said claims under 35 U.S.C. § 101 in the prior office action is hereby withdrawn. However, rejection of claims 46-51 under 35 U.S.C. § 101 is maintained.
4. In light of the amendment made to claim 26, rejection of said claim under 35 U.S.C. 112 second paragraph in the prior office action is hereby withdrawn.

Election by Original Presentation

5. Newly submitted claims 64-74 are directed to an invention that is independent or distinct from the invention originally claimed for the following reasons: claims 64-65 is directed to "a method for providing a user interface ("UI")", claims 66-67 is directed to "a method of generating a user interface for

transacting with a data store having a data model comprising a plurality of entities", claim 68 is directed to "a method for generating a user interface for transacting with a data store", claims 69-71 is directed to "a method for generating a user interface for a client that interacts with a data store", claim 72 is directed to "a computer readable medium storing a computer program for execution by at least one processor, the computer program comprising sets of instructions for (a) analyzing a data model of the data store to produce a description of the structure of the data store ; and (b) distributing the description to a client computer in a platform-independent storage structure", and claims 73-74 is directed to "a method for generating a user interface that interacts with a data store".

Since applicant has received an action on the merits for the originally presented invention, this invention has been constructively elected by original presentation for prosecution on the merits. Accordingly, claim 64-74 withdrawn from consideration as being directed to a non-elected invention. See 37 CFR 1.142(b) and MPEP § 821.03.

Specification

6. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required. Claim 59 in lines 5-10 recites "the first and second users having different roles; and supplying a first description to the first user and a second description to the second user, wherein the first

and second descriptions are different based on the roles of the users". However, the specification fails to provide proper antecedent for said claim limitation.

Claim Rejections - 35 USC § 101

7. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

8. Claims 46-51 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 46-51 is directed to a system. Said system is software per se. Specification only defines "a computer system" – not "system". A server as defined in the abstract of the specification is an application server. Additionally, Figure 2 of the specification clearly defines a "a server communicatively coupled to the database", which is all software per se. Therefore, said system of claim 46-51 is software per se and does not fall within the four statutory categories.

Applicant in the remarks argued that ""Claims 46-51 provide a tangible result through their non-abstract matter. The database must necessarily be non-abstract (i.e., tangible)" (Applicant's remarks, Page 15 last paragraph and Page 16 first paragraph). Applicant is respectfully reminded that claims 46-51 are rejected under 35 U.S.C. 101 as being software per se - not because the subject matter is abstract or not because the claim does not produce concrete results.

Claims 47-51 are rejected 35 U.S.C. 101 because said claims depend on claim 46.

Claim Rejections - 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this

Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

10. Claims 20, 24-27, 31, 35-37, 42-44, 46-50, 52, 53, 56-58, and 63 are rejected under 35 U.S.C. 102(e) as being anticipated by Galea et al., (hereinafter "Galea", U.S. Patent Number 6404445).

As per claim 20, Galea is directed to a method for creating a description of a user interface that transacts with a database having a data model containing a plurality of entities, the description being created using the data model of the database (Galea, Figure 3 and Column 7 Lines 23-41, where it is clear that a data model or GUI database 302 is received by the modeler 304) and teaches the limitations:

a) "classifying the plurality of entities into entity types" (Galea, Column 5 Lines 23-50, i.e., *In one embodiment, the databases contain compiled product or*

server components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations.

*Prerequisite tags may indicate the **GUI element type** such as, for example, single select **list box**, **multi-select list box**, **radio buttons**, **click boxes**, and **input text field**, or images that present the selections options for that domain.*

Optional domain tags determine the run time behavior of each GUI element such as, for example, whether or not to display invalidated options (and subsequently allow single selections overrides), whether or not user interface pages are presented as sequential pages or as grouped elements, whether or not pages/groups for each domain are to be displayed and where they are to be displayed, whether or not the domain is hidden (not included in the user interface), whether or not the domain is displayed as a read only form element (whereby the domain selection may only be made through configuration logic), whether or not to include local or specific domain and domain options representations, whether or not and which corresponding images are to be displayed with each option selection, whether or not and in which form an image element is to be updated with the corresponding image, whether or not and for which corresponding help pages are available for each domain and option selection, whether or not and in what form a status text to offer when navigation to each domain is included, and whether or not the domain is required for final product selection (order inclusion; This disclosure of Galea clearly teaches that data model contained in the GUI database 302 contains a plurality of entity types, that is, the database contains compiled product or server components (also

referred to as domains and each of these domains contains descriptive binding domain elements to their respective GUI representations. Therefore, domains are entities and each entity describes a type of data structure such as various graphical user elements such as radio buttons, click boxes, input text fields etc.);), the classifying comprising:

“determining whether a first entity in the plurality of entities satisfies a first set of conditions; and

classifying the first entity as a first entity type upon determining that the first entity satisfies the first set of conditions” (Galea Figure 8; entities (i.e., user elements) are determined and classified as “*Images*”, “*Navigation Frame*” entities, “*Dynamic Content Frame elements*”; Galea, Column 5 Lines 23-50, i.e., *Prerequisite tags may indicate the GUI element type* such as, for example, *single select list box, multi-select list box, radio buttons, click boxes, and input text field*, or images that present the selections options for that domain); and

b) “creating the description of the user interface based upon the classification of the plurality of entities” (Galea, Column 5 Lines 51-59, i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is downloaded*

to the browser of client 106, 108. **The user interface is dynamically generated based upon the compiled domain tags).**

As per claim 24, Official Note is taken that writing descriptions in XML is notoriously well known in the art. Galea also teaches XML format in Column 8 Lines 25-67 and Column 9 Lines 1-22.

As per claim 25, Galea teaches the limitation:

"wherein the classifying and creating are performed automatically without human assistance" (Galea, Column 6 Lines 20-34, i.e., *The distributed client-side e-commerce service system 100 allows for the **automatic generation** of a user interface for a particular product and/or service model "on-the-fly" such that a web user is always presented with the latest product or service selection choices).*

As per claim 26, Galea teaches the limitation:

"before the classifying, obtaining a current data model of the database, the current data model reflecting any changes to the database up to when the current data model is obtained, wherein a current description of the user interface is created using the current data model of the database" (Galea, Column 6 Lines 20-34, i.e., *The distributed client-side e-commerce service system 100 allows for the **automatic generation** of a user interface for a particular product and/or service model "on-the-fly" such that a web user is always presented with the*

latest product or service selection choices). In Column 10 Lines 44-57, Galea teaches database modeling. From these teachings, it can be inferred that Galea obtains a current data model of the database.

As per claim 27, Galea teaches the limitation:

“before the classifying, receiving a request from a client that the description be created, wherein receiving the request triggers the classifying of entities into entity types” (Galea, Figure 7 and Column 6 Lines 20-34). Galea teaches the client request as show in Figure 7 as well as described in Column 6 Lines 20-34, in an e-commerce system that “*allows for the **automatic generation** of a user interface for a particular product and/or service model “on-the-fly” such that a web user is always presented with the **latest** product or service selection choices.*” It can be inferred that the use and presentation of the product and service selections of the e-commerce taught by Galea can be interpreted as a request for creation of a description of a user interface onto a client system.

Claim 31 is essentially the same as claim 20 except that it set forth the claimed invention as a computer readable medium rather than a method and rejected for the same reasons as applied hereinabove.

As per claim 35, Galea is directed to a method for generating a user interface that transacts with a database having a data model containing a

plurality of entities (Galea, Figure 3 and Column7 Lines 23-41, where it is clear that a data model or GUI database 302 is received by the modeler 304) and teaches the limitations:

"receiving a description of the user interface" (Galea, Column 5 Lines 51-59, i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format*) "the description based upon classification of the plurality of entities into entity types" (Galea, Column 5 Lines 23-50, i.e., *In one embodiment, the databases contain compiled product or server components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations. Prerequisite tags may indicate the **GUI element type** such as, for example, single select **list box**, **multi-select list box**, **radio buttons**, **click boxes**, and **input text field**, or images that present the selections options for that domain. Optional domain tags determine the run time behavior of each GUI element such as, for example, whether or not to display invalidated options (and subsequently allow single selections overrides), whether or not user interface pages are presented as sequential pages or as grouped elements, whether or not pages/groups for each domain are to be displayed and where they are to be displayed, whether or not the domain is hidden (not included in the user interface), whether or not the domain is displayed as a read only form element (whereby the domain selection may only be made through configuration logic),*

whether or not to include local or specific domain and domain options representations, whether or not and which corresponding images are to be displayed with each option selection, whether or not and in which form an image element is to be updated with the corresponding image, whether or not and for which corresponding help pages are available for each domain and option selection, whether or not and in what form a status text to offer when navigation to each domain is included, and whether or not the domain is required for final product selection (order inclusion; This disclosure of Galea clearly teaches that data model contained in the GUI database 302 contains a plurality of entity types, that is, the database contains compiled product or server components (also referred to as domains and each of these domains contains descriptive binding domain elements to their respective GUI representations. Therefore, domains are entities and each entity describes a type of data structure such as various graphical user elements such as radio buttons, click boxes, input text fields etc.)) "wherein the classification comprises classification of a first entity as a first entity type" (Galea Figure 8; entities (i.e., user elements) are determined and classified as "Images", "Navigation Frame" entities, "Dynamic Content Frame elements; Galea, Column 5 Lines 23-50, i.e., *Prerequisite tags may indicate the **GUI element type** such as, for example, single select **list box**, **multi-select list box**, **radio buttons**, **click boxes**, and **input text field**, or images that present the selections options for that domain)* "upon determination that the first entity satisfies a first set of conditions" (Galea Figure 8; entities (i.e., user elements) are determined and classified as "Images", "Navigation Frame" entities, "Dynamic

*Content Frame elements; Galea, Column 5 Lines 23-50, i.e., Prerequisite tags may indicate the **GUI element type** such as, for example, single select **list box**, **multi-select list box**, **radio buttons**, **click boxes**, and **input text field**, or images that present the selections options for that domain) ; and*

*"generating the user interface using the description of the user interface" (Galea, Column 5 Lines 51-59, i.e., A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is downloaded to the browser of client 106, 108. **The user interface is dynamically generated based upon the compiled domain tags**).*

Claim 36 is rejected on the same basis as claim 27.

As per claim 37, Galea teaches the limitation:

"before the receiving: sending preferences for the user interface, the preferences being utilized in creating the description" (Galea, Column 7 Lines 42-47, i.e., A user accesses server 102 via WAN 112 from client 106. When the user makes a selection as to a domain (for example, an online camera store), the compiled interactive decision map 310 for the requested domain is downloaded to client 106. In addition, a client applet is downloaded to client 106. The client

applet downloads the interactive decision map 310 for the particular domain and builds a multi-page graphical user interface (GUI) map by looping through the interactive decision map 310 and collecting array elements for each configuration domain; Galea, Column 11 Lines 65-66, i.e., Window 900 is displayed after a user selects a particular domain for viewing; and Galea, Column 12 Lines 4-6, i.e., After the user accesses a particular domain, the information is downloaded to the client from the server and the client applet downloaded from the server is used to create the display). The domain received from the client by the application server can be interpreted as the preference selected by the user and it is clear that the selected domain will be utilized in creating a description of the user interface as shown in Figure 15.

As per claim 42, Galea is directed to a computer readable medium storing a computer program which when executed by at least one processor generates a user interface that transacts with a database having data model containing a plurality of entities (Galea, Figure 3 and Column 7 Lines 23-41, where it is clear that a data model or GUI database 302 is received by the modeler 304) and teaches the limitations:

"instructions for receiving a description of the user interface, the description being based upon classification of the plurality of entities into entity types wherein the classification comprises classification of a first entity as a first entity type upon determination that the first entity satisfies a first set of

conditions" (Galea, Column 6 Lines 20-34; Column 10 Lines 44-57; Column 5 Lines 51-59); and

"instructions for generating the user interface using the description of the user interface" (Galea, Column 10 Lines 44-59).

As per claim 43, Galea teaches the limitation:

"wherein the description of the user interface is created using a current data model of the database, the current data model reflecting any changes to the database up to when the description is created" (Galea, Column 6 Lines 20-34, i.e., *The distributed client-side e-commerce service system 100 allows for the **automatic generation** of a user interface for a particular product and/or service model "on-the-fly" such that a web user is always presented with the **latest** product or service selection choices*). In Column 10 Lines 44-57, Galea teaches database modeling. From these teachings, it can be inferred that Galea obtains a current data model of the database.

As per claim 44, Galea teaches the limitation:

"instructions for sending preferences for the user interface, the preferences being utilized in creating the description" (Galea, Column 7 Lines 42-47, i.e., *A user accesses server 102 via WAN 112 from client 106. When the user makes a selection as to a domain (for example, an online camera store), the compiled interactive decision map 310 for the requested domain is downloaded*

to client 106. In addition, a client applet is downloaded to client 106. The client applet downloads the interactive decision map 310 for the particular domain and builds a multi-page graphical user interface (GUI) map by looping through the interactive decision map 310 and collecting array elements for each configuration domain; Galea, Column 11 Lines 65-66, i.e., Window 900 is displayed after a user selects a particular domain for viewing; and Galea, Column 12 Lines 4-6, i.e., After the user accesses a particular domain, the information is downloaded to the client from the server and the client applet downloaded from the server is used to create the display). The domain received from the client by the application server can be interpreted as the preference selected by the user and it is clear that the selected domain will be utilized in creating a description of the user interface as shown in Figure 15.

As per claim 46, Galea is directed to a system and teaches the limitations:

“a database having a data model and containing a plurality of entities”

(Galea, Figure 3 and Column 7 Lines 23-41, where it is clear that a data model or GUI database 302 is received by the modeler 304) ; and

*“a server communicatively coupled to the database” (Galea, Column 5 Lines 11-15, i.e., Figure 1 is an illustration of one embodiment for a distributed client-side e-commerce service system 100. Referring to Figure 1, **server 100** is coupled to mass storage 104. Server 102 and mass storage device 104 are coupled via wide area network (WAN); and Galea, Figure 3 : 104 (which is mass storage device) which comprises product & GUI data 302) “for creating a*

description of a user interface that interacts with the database" (Galea, Column 5 Lines 51-59, i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is downloaded to the browser of client 106, 108. **The user interface is dynamically generated based upon the compiled domain tags***), "the description being based upon classification of the plurality of entities into entity types" (Galea, Column 5 Lines 23-50, i.e., *In one embodiment, the databases contain compiled product or server components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations. Prerequisite tags may indicate the **GUI element type** such as, for example, single select **list box**, **multi-select list box**, **radio buttons**, **click boxes**, and **input text field**, or images that present the selections options for that domain. Optional domain tags determine the run time behavior of each GUI element such as, for example, whether or not to display invalidated options (and subsequently allow single selections overrides), whether or not user interface pages are presented as sequential pages or as grouped elements, whether or not pages/groups for each domain are to be displayed and where they are to be displayed, whether or not the domain is hidden (not included in the user interface), whether or not the domain is displayed as a read only form element (whereby the domain selection may only be made through configuration*

*logic), whether or not to include local or specific domain and domain options representations, whether or not and which corresponding images are to be displayed with each option selection, whether or not and in which form an image element is to be updated with the corresponding image, whether or not and for which corresponding help pages are available for each domain and option selection, whether or not and in what form a status text to offer when navigation to each domain is included, and whether or not the domain is required for final product selection (order inclusion; This disclosure of Galea clearly teaches that data model contained in the GUI database 302 contains a plurality of entity types, that is, the database contains compiled product or server components (also referred to as domains and each of these domains contains descriptive binding domain elements to their respective GUI representations. Therefore, domains are entities and each entity describes a type of data structure such as various graphical user elements such as radio buttons, click boxes, input text fields etc.) "wherein the classification comprises classification of a first entry as a first entity type upon determination that the first entity satisfies a first set of conditions" (Galea Figure 8; entities (i.e., user elements) are determined and classified as "Images", "Navigation Frame" entities, "Dynamic Content Frame elements; Galea, Column 5 Lines 23-50, i.e., Prerequisite tags may indicate the **GUI element type** such as, for example, **single select list box**, **multi-select list box**, **radio buttons**, **click boxes**, and **input text field**, or images that present the selections options for that domain).*

As per claim 47, Galea teaches the limitation:

“wherein the server obtains a current model of the database, the current data model reflecting any changes to the database up to when the current model is obtained, a current description of the user interface being created using the current data model of the database” (Galea, Column 6 Lines 20-34, i.e., *The distributed client-side e-commerce service system 100 allows for the **automatic generation** of a user interface for a particular product and/or service model “on-the-fly” such that a web user is always presented with the **latest** product or service selection choices*). In Column 10 Lines 44-57, Galea teaches database modeling. From these teachings, it can be inferred that Galea obtains a current data model of the database.

As Claim 48, Galea teaches the limitation:

“wherein the server is in persistent communication with the database” (Galea, Figure 3).

As per claim 49 Galea teaches the limitations:

“wherein the server is communicatively coupled to a first client via a network” (Galea Figure 1) and “distributes the created description to the first client for enabling the first client to generate the user interface” (Galea, Column 5 Lines 51-59 and Column 6 Lines 20-34).

As per claim 50, Galea teaches the limitation:

"wherein the server provides the first client an only point of access to the database" (Galea, Figure 1 and Figure 3).

As per claim 52, Galea teaches is directed to a computer and teaches the limitations:

a) "a description of a data store" (Galea, Galea, Column 5 Lines 51-59, i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is downloaded to the browser of client 106, 108. The user interface is dynamically generated based upon the compiled domain tags*)

b) "a browser" (Galea, Figure 6, i.e., *Browser Application*); and

c) "an application for generating user interface elements by using said description and by retrieving a data set from said data store to populate at least one user interface element, said user interface elements for displaying in said browser" (Galea, Galea, Column 5 Lines 51-59 i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is **downloaded to the***

browser of client 106, 108. *The user interface is dynamically generated based upon the compiled domain tags) and "at least one user interface element for receiving queries for the data store" (Galea, Figure 11, i.e., SEARCH 1126 and Galea, Column 12 Lines 16-27, i.e., FIG. 11 illustrates an exemplary feature selection window 1100. Window 1100 includes a feature selection pop up window 1102 containing a number of feature selection pull down boxes 1104-1124. In addition, window 1100 also includes a **search box 1126** and a browse box 1128. After a domain file has been down loaded from the server to the client, the user may make selections of the various products from any of the pull down boxes (1104-1124). For example, the user may pull down picture quality box 1104 to give the selections shown at 1144. In this example, picture qualities may be selected based on poor, decent, good, very good, excellent or any combination deemed suitable).*

As per claim 53, Galea teaches the limitation:

"comprising a storage for storing the description, the browser and the application" (Galea Figures 1, 3, and 6).

As per claim 56, Galea teaches the limitation:

"wherein said browser is a web browser" (Galea Figure 6), "wherein said application is a distributed application running on said web browser" (Galea, Figure 3).

As per claim 57, Official Note is taken that distributing an application, as an applet is notorious well known in the art, as can be seen Java applets on millions of web sites. Galea also teaches a Java runtime applet on client 106 in column 6 lines 60-62.

As per claim 58, Galea teaches the limitations:

a) "receiving a first request for a first user interface to transact with a first data store" (Galea Column 5 Lines 51-55, i.e., *A user access a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interfaces tags, is compiled into a secure, binary compressed file format.* And Galea, Column 5 Lines 10-22, i.e., *FIG. 1 is an illustration of one embodiment for a distributed client-side e-commerce service system 100. Referring to FIG. 1, server 102 is coupled to mass storage device 104. Server 102 and mass storage device 104 are coupled via wide area network (WAN) 112 to a variety of clients 106 and 108. Wide area network 112 may be coupled to any of a variety of clients 106 and 108. In one embodiment, mass storage device 104 contains a product component and graphical user interface (GUI) database and an interactive decision map database. In an alternate embodiment, each of the databases may be contained in a separate mass storage devices 104*);

b) supplying a first description to generate the first user interface" (Galea, Column 4 Lines 1-5, i.e., *the constraint-based configuration file is mapped into a*

plurality of pages at a client, and a page of **the plurality of page** is updated directly from the constraint-based configuration file; Column 5 Lines 19-22, i.e., *In an alternate embodiment, each of the databases may be contained in a separate mass storage devices 104*; Column 5 Lines 23-26, i.e. *In one embodiment, the databases contain compiled product or service components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations; Figure 12 and Figure 13; Column 12 Lines 28-50, which explains Figure 12 and Figure 13; Also note Figure 1 wherein Client 106 (first client) and Client 108 (second client) can access the server for different user interfaces));*

c) "receiving a second request for a second user interface to transact with a second data store" (Galea Column 5 Lines 51-55, i.e., *A user access a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interfaces tags , is compiled into a secure, binary compressed file format. And Galea, Column 5 Lines 10-22, i.e., FIG. 1 is an illustration of one embodiment for a distributed client-side e-commerce service system 100. Referring to FIG. 1, server 102 is coupled to mass storage device 104. Server 102 and mass storage device 104 are coupled via wide area network (WAN) 112 to a variety of clients 106 and 108. Wide area network 112 may be coupled to any of a variety of clients 106 and 108. In one embodiment, mass storage device 104 contains a product component and graphical user interface (GUI) database and an interactive decision map*

*database. In an alternate embodiment, each of the databases may be contained in a separate mass storage devices 104; Figure 12 and Figure 13; Column 12 Lines 28-50, which explains Figure 12 and Figure 13; Also note Figure 1 wherein Client 106 (first client) and Client 108 (second client) can access the server for different user interfaces); Particular note Galea's disclosure which states that **"In an alternative embodiment, each of the databases may be contained in a separate mass storage devices 104,** which clearly teaches that there are more than one data stores which different clients (users) could access;*

*and d) supplying a second description to generate the second user interface, wherein said first and second descriptions differ" (Galea, Column 4 Lines 1-5, i.e., the constraint-based configuration file is mapped into a **plurality of pages** at a client, and a page of the **plurality of page** is updated directly from the constraint-based configuration file; Column 5 Lines 19-22, i.e., In an alternate embodiment, **each of the databases may be contained in a separate mass storage devices 104;** Column 5 Lines 23-26, i.e. In one embodiment, the databases contain compiled product or service components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations; Figure 12 and Figure 13; Column 12 Lines 28-50, which explains Figure 12 and Figure 13; Also note Figure 1 wherein Client 106 (first client) and Client 108 (second client) can access the server for different user interfaces).*

As per claim 63 Galea teaches the limitation:

"wherein the first and second data stores are the different data stores, wherein the first request is received from a first user while the second request is received from a second user different than the first, wherein the description is supplied to the first user while the second description is supplied to the second user" (Galea, Column 4 Lines 1-5, i.e., *the constraint-based configuration file is mapped into a **plurality of pages** at a client, and a page of **the plurality of page** is updated directly from the constraint-based configuration file*; Column 5 Lines 19-22, i.e., *In an alternate embodiment, **each of the databases may be contained in a separate mass storage devices 104***; Column 5 Lines 23-26, i.e. *In one embodiment, the databases contain compiled product or service components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations*; Figure 12 and Figure 13; Column 12 Lines 28-50, which explains Figure 12 and Figure 13; Also note Figure 1 wherein *Client 106* (first client) and *Client 108* (second client) can access the server for different user interfaces).

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

12. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

13. Claims 59-62 are rejected under 35 U.S.C. 103(a) as being unpatentable over Galea in view of Russell et al., (hereinafter "Russell", U.S. Patent Number 7076784).

As per claim 59, Galea is directed to a method for providing descriptions of user interfaces (Galea, Column 5 Lines 51-59, i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format*) and teaches the limitations:

(a) "receiving a request from a first user" (Figure 1 wherein by way of *Client 106* (first client) and *Client 108* (second client), different users can access the server for different user interfaces);

(b) "receiving a second request from a second user" (Figure 1 wherein by way of *Client 106* (first client) and *Client 108* (second client), different users can access the server for different user interfaces), (the first and second users having different roles);

(c) "supplying a first description to the first user and a second description to the second user" (Figure 1 wherein by way of *Client 106* (first client) and *Client 108* (second client), different users can access the server for different user interfaces; Galea, Column 4 Lines 1-5, i.e., *the constraint-based configuration file is mapped into a plurality of pages at a client, and a page of the plurality of page is updated directly from the constraint-based configuration file*; Column 5 Lines 19-22, i.e., *In an alternate embodiment, each of the databases may be contained in a separate mass storage devices 104*; Column 5 Lines 23-26, i.e. *In one embodiment, the databases contain compiled product or service components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations*; Figure 12 and Figure 13; Column 12 Lines 28-50, which explains Figure 12 and Figure 13; Also note Figure 1 wherein *Client 106* (first client) and *Client 108* (second client) can access the server for different user interfaces) (wherein the first and second descriptions are different based on the roles of the users).

Galea does not explicitly teach the limitations: "the first and second users having different roles" and "wherein the first and second descriptions are different based on the roles of the users".

On the other hand, Russell teaches the limitations:

“the first and second users having different roles” and “wherein the first and second descriptions are different based on the roles of the users” (Russell, Column 44 Lines 33-45, i.e., *With reference again to FIG. 18, the developer assigns access privileges of a role to the package by adding the role to the roles folder of the package as just discussed. As described in the Authorization Checking section below, this allows user threads operating under the role to have access to the package. The developer can additionally assign access privileges of certain subsets of the roles having access privileges to the package to particular components and component interfaces using the Transaction Server Explorer. This allows the developer to declaratively configure different logical classes (i.e., roles) to have different levels of access to particular processing services of the server application. Note that Russell teaches providing different software packages based on user roles.*

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify the method of Galea to add the feature of providing different software packages to different users based on user roles, as taught by Russell so that, in the resultant method, different users (a first user and a second user) would be supplied with different descriptions based on the roles of the first and second users. One would have been motivated to do so in order to provide management of access control with respect to data objects (Russell, Column 9 Lines 51-56).

As per claim 60, Galea in view of Russell teaches the limitations:

"wherein the first and second user interfaces comprises at least two interface elements for facilitating data transactions" (see Galea, Figure 12 and 13, wherein more than two interface elements are displayed), "wherein said first interface comprises at least one or more user interface elements than said second user interface" (Galea Figure 1 describes 2 clients and Galea Figures 12 and 13 describes a plurality of interface elements on each generated user interfaces. Depending on domains, numbers of user interface elements are different. As such, one interface always has more interface elements than other interface).

As per claim 61, Galea in view of Russell teaches the limitation:

"wherein first and second user interfaces are displayed in an application running different computers" (Galea, Figure 1, *Client 106* (first client) and *Client 108* (second client)).

As per claim 62, Galea in view of Russell teaches the limitation:

"wherein the application is web browser" (Galea, Galea, Column 5 Lines 51-59 i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is downloaded to the browser of client 106, 108.*).

14. Claim 21 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Galea in view of Roth (U.S. Patent Number 6564218).

As per claim 21, Galea teaches the limitations:

“each entity in the data model describes a type of data object associated with the database” (Galea, Column 10 Lines 44-57, i.e., *FIG. 7 is a flow diagram of one embodiment for **modeling data** for a client-side e-commerce system 100. Initially at processing block 702, the data configuration domain for a particular e-commerce site is defined. In one embodiment, system 100 models data by defining and binding relations between data items. At processing block 703, graphical user interface (GUI) tags are entered and dynamically generated. The GUI tags are used to define relations between the data and the graphical representation of the data. The graphical user interface (GUI) tags, together with configuration domain relations generated at processing block 702, are modeled into an extended markup language (XML) configuration file. The XML configuration file is saved in product in GUI data 302*); and

“the classifying produces the first entity type for a first group of data objects and a second entity type for a second group of data objects” (Galea Figure 8; and Column 11 Lines 48-63, i.e., *Fields are input that indicate the user interface elements types and optional domain tags which determine the run time behavior of each of the user interface elements*).

Galea does not explicitly teach the limitation: "the data objects in the first group of data objects being updated in the database more frequently than the data objects in the second group of data objects".

On the other hand, Roth teaches the limitation:

"the data objects in the first group of data objects being updated in the database more frequently than the data objects in the second group of data objects" (Roth, Column 4 Lines 54-57, i.e., *Advantageously, said one or more supersets to be used for said validity check are selected from said specified supersets on the basis of a priori knowledge of supersets least likely to be updated*). Roth teaches supersets of data objects and some of said supersets are more frequently update while others are less frequently updated.

At the invention was made, it would have been obvious to a person of ordinary skill in the to add the feature of updating data objects/entities, of which some are more frequently updated and some are less frequently updated, as taught by Roth, to the method of Galea so that, in the resultant method, the classifying would produce the first entity type for a first group of data objects and a second entity type for a second group of data objects, the data objects in the first group of data objects being updated in the database more frequently than the data objects in the second group of data objects. One would have been motivated to do so in order to improve overall speed of retrieval/updates of digital information (Roth, Column 4 Lines 39-43).

As per claim 22, Galea in view of Roth teaches the limitations:

"wherein the first entity type is a Main entity type and the second entity type is an Enumeration type" (Roth, Column 8 Lines 65 through Column 9 Line 5, i.e., *As **changes typically affect some parameters with a higher probability than others**, the version comparison can be optimized by starting the comparison using the parameter with **the least probability of change**, e.g. **starting with the parameter "country"**, provided that data are changed on a country by country basis*).

15. Claim 23, 32, and 51 are rejected under 35 U.S.C. 103(a) as being unpatentable over Galea in view of Sexton et al., (hereinafter "Sexton") (U.S. Patent Number 7093263).

As per claim 23, Galea does not explicitly teach the limitation: "wherein the description is a generic description configured to be interpreted in different platforms or operating environments".

On the other hand, Sexton teaches the limitation:

"wherein the description is a generic description configured to be interpreted in different platforms or operating environments" (Sexton, Column 3 Lines 40-52, i.e., *In addition, instructions can automatically be generated for getting and setting values in the object in the platform-independent format, thereby diminishing the reliance on manually coding the operations. As a result, errors are reduced and the code is more maintainable*).

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to add the feature of employing generic

descriptions/instructions which can be interpreted in different operating environments, as taught by Sexton, to the method of Galea so that, in the resultant method, the description would be a generic description which is to be interpreted in different platforms. One would have been motivated to do so in order to make the code/instructions/descriptions portable, which is a well-known practice in the art, such as Java code.

Claim 32 is rejected on the same basis as claim 23.

As per claim 51, Galea in view of Sexton teaches the limitations:

“wherein the server is communicatively coupled, via the network, to a second client” (Galea Figure 1 in view of Sexton Figure 1) “having a different platform or operating environment than the first client” (Column 9 Lines 3-6, i.e., *For example, if a platform does not have a 4-byte primitive integer (e.g., on 64-bit machines such as **Cray YP-1**),* and “distributes the created description to the second client for enabling the second client to generate the user interface” (Galea, Column 5 Lines 51-59 and Column 6 Lines 20-34).

16. Claim 38 is rejected under 35 U.S.C. 103(a) as being unpatentable over Galea in view of Mashayekhi (U.S. Patent Number 5818936).

As per claim 38, Galea does not explicitly teach the limitation: “before the receiving: sending authentication information”.

On the other hand, Mashayekhi teaches the limitation:

“before the receiving: sending authentication information” (Mashayekhi, Column 7 Lines 10-30, i.e., *when a user 201 attempts to access a particular application program, such as a local application 240 or network-based application program 236, the particular application program requires that the user be authenticated prior to accessing its processes or data*).

At the time the invention was made, it would have been obvious to a person of ordinary skill to add the feature of authenticating a user access, as taught by Mashayekhi, to the method of Galea so that the resultant method would comprise sending authentication information. One would have been motivated to do so in order to provide security to the database, which is well known in the art.

Response to Arguments

17. Applicant's arguments filed on **November 2, 2007** have been considered but are not persuasive.

With respect to rejection of claims 34 and 45-51 under 35 U.S.C. 101 in the prior office action, Applicant in the remarks argued that “Claims 46-51 provide a tangible result through their non-abstract matter. The database must necessarily be non-abstract (i.e., tangible)” (Applicant's remarks, Page 15 last paragraph and Page 16 first paragraph). Applicant is respectfully reminded that claims 46-51 are rejected under 35 U.S.C. 101 as being software per se - not

because the subject matter is abstract or not because the claim does not produce concrete results.

Applicant also argued that "*For at least two reasons, Applicants respectfully submit that Galea does not disclose, teach, or even suggest the method of claim 20. First, Galea does not disclose, teach, or even suggest classifying entities into entity types*" (Applicant's argument, page 16 last paragraph). Applicant also argued that "*However, nowhere in the cited section or anywhere else does Galea disclose classifying entities into entity types*" (Applicant's argument, page 17 first paragraph).

Examiner respectfully disagrees all of the allegations as argued. Examiner, in his previous office action, gave detail explanation of claimed limitation and pointed out exact locations in the cited prior art. Examiner is entitled to give claim limitations their broadest reasonable interpretation in light of the specification. See MPEP 2111 [R-1] Interpretation of Claims-Broadest Reasonable Interpretation.

During patent examination, the pending claims must be 'given the broadest reasonable interpretation consistent with the specification.' Applicant always has the opportunity to amend the claims during prosecution and broad interpretation by the examiner reduces the possibility that the claim, once issued, will be interpreted more broadly than is justified. In re Prater, 162 USPQ 541,550-51 (CCPA 1969).

In response it is pointed out that Galea teaches classifying entities into entity types in Column 5 Lines 23-50 as *In one embodiment, the databases*

*contain compiled product or server components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations. Prerequisite tags may indicate the **GUI element type** such as, for example, single select list box, multi-select list box, radio buttons, click boxes, and input text field, or images that present the selections options for that domain. Note that GUI elements are entities and said GUI elements are classified into entity types such as single select list box, multi-select list box, radio buttons, click boxes, and input text field, or images.*

Applicant additionally argued that "Second, Galea dose not disclose, teach, or even suggest classifying an entity upon determining that the entity satisfies the set of conditions" (Applicant's argument, page 17 second paragraph). Applicant also argued that "Galea does not disclose, teach, or even suggest classifying an entity upon determining that the entity satisfies a set of conditions" (Applicant's argtument, page 17, second paragraph).

In response, it is pointed Galea teaches *classifying an entity upon determining that the entity satisfies the set of conditions* as follows: a) "classifying the plurality of entities into entity types" (Galea, Column 5 Lines 23-50, i.e., *In one embodiment, the databases contain compiled product or server components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations. Prerequisite tags may indicate the **GUI element type** such as, for example, single select **list box**, **multi-select list box**, **radio buttons**, **click boxes**, and **input text field**, or images that present the selections options for that domain;* "determining whether a first entity in the plurality of entities satisfies a first set of conditions" (Galea Figure 8;

entities (i.e., user elements) are determined and classified as "*Images*", "*Navigation Frame*" entities, "*Dynamic Content Frame elements*"; Galea, Column 5 Lines 23-50, i.e., *Prerequisite tags may indicate the **GUI element type** such as, for example, single select **list box**, **multi-select list box**, **radio buttons**, **click boxes**, and **input text field**, or images that present the selections options for that domain).*

Applicant argued that "*As claims 21-27 are dependent directly or indirectly on claim 20, Applicant respectfully submit that claims 21-27 are patentable over Galea for at least the reasons discussed above for claim 20*" (Applicant's argument, page 17, third paragraph).

In response, it is pointed out that since claim 20 is not patentable over Galea, all and any dependent claims of claim 20 are not patentable.

Applicant also argued that "*Accordingly, for reasons similar to those stated above for claim 20, Applicant respectfully submit that Galea does not disclose, teach, or even suggest classifying entities into entity types. Also, for reasons similar to those stated above for claim 20, Applicant respectfully submit that Galea does not disclose, or teach, or even suggest classifying an entity upon determining the entity satisfies a set of conditions*" (Applicant's argument, page 18 second paragraph).

In response, it is pointed out that said limitations are clearly taught by Galea as discussed above with respect to claim 20.

Applicant argued that "*Applicant respectfully submit that Galea does not render claim 31 unpatentable. As claim 32 and 33 are dependent directly on claim 31, Applicants respectfully submit that claims 32 and 33 are patentable*

over Galea for at least the reasons discussed above for claim 31 (Applicant's argument, page 13 last paragraph).

In response, it is pointed out that since claim 31 is not patentable over Galea, all and any dependent claims of claim 31 are not patentable.

Applicant argued that *"Accordingly, for reasons similar to those stated above for claim 20, Applicant respectfully submit that Galea does not disclose, teach, or even suggest receiving a description of a user interface that is based upon classification of entities into entity types. Also, reasons similar to those stated above for claim 20, Applicant respectfully submit that Galea dose not disclose, teach, or even suggest classification of an entity as an entity type upon determination that the entity satisfies a set of conditions"* (Applicant's argument, page 19, second paragraph).

In response, it is pointed out that said limitations are clearly taught by Galea as discussed above with respect to claim 20.

Applicant argued that *"Applicant respectfully submit that Galea dose not render claim 35 unpatentable. As claim 36-38 are dependent directly on claim 35, Applicants respectfully submit that claims 36-38 are patentable over Galea for at least the reasons discussed above for claim 35* (Applicant's argument, page 19 last paragraph and page 20 first paragraph).

In response, it is pointed out that since claim 35 is not patentable over Galea, all and any dependent claims of claim 35 are not patentable.

Applicant argued that *"Galea does not disclose, teach, or even suggest a computer program that receives a description where the description is based on*

classifying the entities into entity types" (Applicant's argument page 20, last paragraph).

In response, it is pointed out that said limitations are clearly taught by Galea as discussed above with respect to claim 20.

Applicant also argued that *"nowhere in the cited section or anywhere else dose Galea disclose, teach, or suggest receiving a description which is based on the classification of entities into entity types"* (Applicant's argument, page 20 last paragraph and page 21 first paragraph). Applicant also argued that *"Accordingly, for reasons similar to those stated above for claim 20, Applicant respectfully submit that Galea does not disclose, teach, or even suggest creating a description based upon classification of entities into entity types. Also, reasons similar to those stated above for claim 20, Applicant respectfully submit that Galea dose not disclose, teach, or even suggest classification of an entity upon determination that the entity satisfies a set of conditions"* (Applicant's argument, page 21 last paragraph and page 22 first paragraph).

In response, it is pointed out that said limitations are clearly taught by Galea as discussed above with respect to claim 20.

Applicant argued that *"Applicant respectfully submit that Galea dose not render claim 46 unpatentable. As claim 47-51 are dependent directly on claim 35, Applicants respectfully submit that claims 47-51 are patentable over Galea for at least the reasons discussed above for claim 46"* (Applicant's argument, page 22 second paragraph).

In response, it is pointed out that since claim 46 is not patentable over Galea, all and any dependent claims of claim 46 are not patentable.

Referring to claim 52, Applicant argued that "*Galea dose not disclose, teach, or even suggest a computer that includes an application for generating a use interface element receiving queries on a data store*" (Applicant's argument, page 22 last paragraph).

In response, it is pointed out that Galea teaches *a computer that includes an application for generating a use interface element receiving queries on a data store* as follows: "an application for generating user interface elements by using said description and by retrieving a data set from said data store to populate at least one user interface element, said user interface elements for displaying in said browser" (Galea, Galea, Column 5 Lines 51-59 i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is **downloaded to the browser of client 106, 108.** The user interface is dynamically generated based upon the compiled domain tags) and "at least one user interface element for receiving queries for the data store" (Galea, Figure 11, i.e., *SEARCH 1126* and Galea, Column 12 Lines 16-27, i.e., *FIG. 11 illustrates an exemplary feature selection window 1100. Window 1100 includes a feature selection pop up**

*window 1102 containing a number of feature selection pull down boxes 1104-1124. In addition, window 1100 also includes a **search box 1126** and a browse box 1128. After a domain file has been down loaded from the server to the client, the user may make selections of the various products from any of the pull down boxes (1104-1124). For example, the user may pull down picture quality box 1104 to give the selections shown at 1144. In this example, picture qualities may be selected based on poor, decent, good, very good, excellent or any combination deemed suitable).*

Applicant argued that "Applicant respectfully submit that Galea dose not render claim 52 unpatentable. As claim 53-57 are dependent directly on claim 35, Applicants respectfully submit that claims 53-57 are patentable over Galea for at least the reasons discussed above for claim 52 (Applicant's argument, page 22 second paragraph).

In response, it is pointed out that since claim 52 is not patentable over Galea, all and any dependent claims of claim 52 are not patentable.

Applicant argued that "Applicants respectfully submit that Galea dose not disclose, teach, or even suggest that the method of claim 58. For instance, Galea does not disclose supplying a user interface for transacting with a data store". (Applicant's argument, page 24 first paragraph).

*In response, it is pointed out that Galea teaches supplying a user interface for transacting a data store as follows: a) "receiving a first request for a first user interface to transact with a first data store" (Galea Column 5 Lines 51-55, i.e., A **user access a server through client 106, 108 via wide area network 112 in order***

*to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interfaces tags, is compiled into a secure, binary compressed file format. And Galea, Column 5 Lines 10-22, i.e., FIG. 1 is an illustration of one embodiment for a distributed client-side e-commerce service system 100. Referring to FIG. 1, server 102 is coupled to **mass storage device 104**. Server 102 and mass storage device 104 are coupled via wide area network (WAN) 112 to a variety of clients 106 and 108. Wide area network 112 may be coupled to any of a variety of clients 106 and 108. In one embodiment, mass storage device 104 contains a product component and graphical user interface (GUI) database and an interactive decision map database. **In an alternate embodiment, each of the databases may be contained in a separate mass storage devices 104).***

Applicant argued that "Applicant respectfully submit that Galea dose not render claim 58 unpatentable. As claim 63 is dependent directly on claim 58, Applicants respectfully submit that claim 63 is patentable over Galea for at least the reasons discussed above for claim 58 (Applicant's argument, page 24 second paragraph).

In response, it is pointed out that since claim 58 is not patentable over Galea, all and any dependent claims of claim 58 are not patentable.

Applicant also argued that "Galea does not disclose, teach, or suggest supplying different descriptions of a data store to different users based on the roles of the users" (Applicant's argument, page 25 second paragraph).

In response, it is pointed out that said argument is made based on amendments made to claim 59 and is moot in view of new ground(s) of rejection.

Applicant argued that "*Applicant respectfully submit that Galea dose not render claim 59 unpatentable. As claims 60-62 is dependent directly on claim 58, Applicants respectfully submit that claims 60-62 is patentable over Galea for at least the reasons discussed above for claim 59* (Applicant's argument, page 25 third paragraph).

In response, it is pointed out that since claim 59 is not patentable over Galea, all and any dependent claims of claim 59 are not patentable.

With respect to Applicant's arguments regarding newly added claims (64-74), said arguments are moot because newly added claims 64-74 are with drawn from consideration by virtue of the election by original presentation, said arguments are moot.

In view of the above, the examiner contends that all limitations as recited in the claims have been addressed in this Action. For the above reasons, Examiner believed that rejection of the last Office action was proper

Conclusion

18. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the date of this final action.

Contact Information

19. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis Myint whose telephone number is (571) 272-5629. The examiner can normally be reached on 8:30AM-5:30PM Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Breene can be reached on (571) 272-4107. The fax phone number for the organization where this application or proceeding is assigned is 571-273-5629.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

[Handwritten Signature]
Sean M. K. Fournier
Primary Examiner
TC 2162

[Handwritten Signature]
Dennis Myint
Examiner
AU-2162